The Renegotiable Variable Bit Rate Service

Silvia Giordano <Silvia.Giordano@epfl.ch>[1],
Jean-Yves Le Boudec <Leboudec@epfl.ch>[1]
[1]EPFL ICA, Lausanne, Switzerland

Abstract: *A shaper is a system that stores incoming bits in a buffer and delivers them as early as possible, while forcing the output to be constrained with a given arrival curve. A shaper is time invariant if the traffic constraint is defined by a fixed arrival curve; it is time varying if the condition on the output is given by a time varying traffic contract. This occurs, for example, with renegotiable variable bit rate (RVBR) services. We focus on the class of time varying shapers called time varying leaky bucket shapers; such shapers are defined by a fixed numbers of leaky buckets, whose parameters (rate and bucket size) are changed at specific transition moments. We assume that the bucket levels are kept unchanged at those transition moments ("no reset" assumption). Our main finding is an input-output characterisation for this class of time varying shapers. Then we apply it to the tradeoff in optimising the RVBR service, assuming that a perfect prediction of future traffic can be made. We provide an algorithm that solves the problem of finding, at any renegotiation, the parameters for a RVBR service when the knowledge of the input traffic is limited to the next interval (local optimisation problem). We illustrate the impact of the "no-reset" assumption by analyzing on some examples the losses that occur when the source chooses the opposite approach, namely, the "reset" approach.*
Keywords: Shaping system, renegotiation, VBR parameters, resources optimisation, RSVP.

1 Introduction

We consider the Renegotiable Variable Bit Rate (RVBR) service, defined as a variable bit rate service whose parameters are changed at periodic renegotiation moments. An example for this service is the Integrated Service of the IETF with the Resource reSerVation Protocol (RSVP), where the negotiated contract may be modified periodically [2]. A flow using the RVBR service is constrained by two leaky buckets: one defines the peak rate, the other defines the sustainable rate and the burst tolerance. We consider a basic scenario where a fresh input traffic is shaped in order to satisfy the leaky bucket constraints. Shaping is assumed to be done using an optimal shaper, with a limited buffer size X [3]. The input traffic may be generated by one source, or it may be an aggregate of sources, in which case the shaper models a service multiplexer. Using VBR in a shaper may be advantageous in all cases where the input traffic is bursty and the network is able to achieve a statistical multiplexing gain on many such input flows [4].

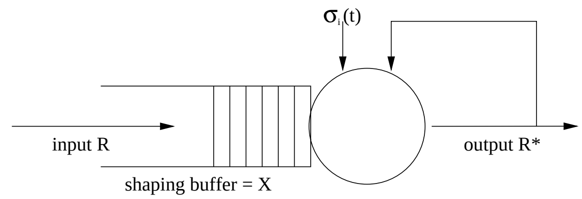


Figure 1: Reference Model for a time varying leaky-bucket shaper. The traffic shaping at time $t \in I_i$ is done at source according to the service curve σ_i valid in I_i.

In our model scenario, the RVBR parameters are renegotiated periodically; at every renegotiation, there is a tradeoff to be made between the various parameters, which define the two leaky buckets in the next interval. Our primary goal in this paper is to analyse this tradeoff. In particular, we propose a method to select, for the next interval, the parameters that minimise a given linear cost function.

Time Varying Shapers

We analyse the RVBR service using a special class of time varying shaper systems, which we call the *time varying leaky-bucket shapers.* This is defined by a fixed number J of leaky bucket specifications with bucket rate r^j and bucket depth b^j, where $j = 1, \ldots, J$ and a shaping buffer of fixed capacity X. At specified time instants t_i, $i = 0, 1, 2, ..$, the parameters of the leaky buckets are modified and $I_i = (t_i, t_{i+1}]$ represents the i-th interval.

Inside each interval the system does not change. The parameters of the j-th leaky buckets valid in the interval I_i are indicated by (r_i^j, b_i^j). The combination of those parameters takes the form of the shaping function σ_i in I_i, defined as

$$\sigma_i(u) = \min_{1 \leq j \leq J}\{\sigma_i^j(u)\} = \min_{1 \leq j \leq J}\{r_i^j \cdot u + b_i^j\}$$

A time varying leaky-bucket shaper is completely defined by the number J of leaky buckets; the time instants t_i at which the parameters changes; the buckets parameters (r_i^j, b_i^j), for each j and each interval I_i; the fixed shaping buffer capacity X.

We call *input traffic function* the function $R(t) : \mathbb{R}^+ \rightarrow \mathbb{R}^+$ that represents the amount of traffic that has entered in the system in time interval $[0, t]$. R is the traffic before the shaping. $R^*(t)$ is the *output function* that represents the number of bytes seen on the output flow in time interval $[0, t]$. R^* is the traffic after the shaping. We assume to know the input traffic $R(t)$ expected in the future either because pre-recorded or by means of an exact prediction function. However the traffic prediction is not the focus of this paper. We further assume that at time $t_0 = 0$ the system is idle ($R(0) = 0$).

To define the time varying leaky-bucket shapers at the transient times between two adjacent intervals we could take two opposite approaches: either we reset all the buckets and restart in the next interval from zero initial conditions ("reset" approach), or we keep the level of the buckets and restart from that level at the next interval ("no reset" approach). If we take the first approach, the time varying leaky-bucket shaper can be reduced to a sequence of independent shapers and studied as in [5], [6]. Here we adopt the second approach. There are two reasons for this. First, in the special case where the time varying leaky-bucket is constant, we should find a system identical to the ordinary, time invariant, leaky bucket shaper [5], [6]. In other words, this is true only with the second approach. Second, the "no reset" approach is in line with the Dynamic Generic Cell Rate Algorithm (DGCRA) used to specify conformance at the UNI for the available bit rate (ABR) service of ATM [7], [8]. We examine later in the paper the practical implication of the "no-reset" approach (Section 4).

Our class of time varying shapers is a special case of the general concept of time varying shapers, defined in [9]. A general time varying shaper can be defined as follows. Given a function of two time variables $W(s, t)$, the time varying shaper forces the output $R^*(t)$ to satisfy the condition

$$R^*(t) \leq R^*(s) + W(s, t)$$

for all $s \leq t$, possibly at the expense of buffering some data. This condition can be expressed using the min-plus linear operator associated to W and defined as the mapping $S \rightarrow S \cdot W$ with $(S \cdot W)(t) = \inf_s\{S(s) + W(s, t)\}$. The shaper is an optimal shaper if it maximises its output among all possible shapers [9]. A time invariant shaper is a special case; it corresponds to $W(s, t) = \sigma(t - s)$, where σ is the shaping curve.

General results of min-plus algebra say that the input-output characterisation of a time-varying shaper is given by

$$R^* = R \cdot \bar{W}$$

where function R is the input, R^* the output and $\bar{W}$ is the sub-additive *closure* of W [10, 11]. Another, equivalent, formulation is:

$$R^*(t) = \inf\{R(t), (R \cdot W)(t), (R \cdot W \cdot W)(t), \dots\} \tag{1}$$

Our class of time varying shapers fits in that general framework. It can be easily shown that a time varying leaky bucket shaper corresponds to

$$W(s, t) = \min_{1 \leq j \leq J}\{\int_s^t r_j(u)du \quad + b_j(t)\} \tag{2}$$

with $r_j(t)$ and $b_j(t)$ defined as the instantaneous bucket rate and depth at time t, namely $r_j(t) = r_j^i$ and $b_j(t) = b_j^i$ for the index i such that $t_i < t \leq t_{i+1}$.

In this paper, we want to obtain the input-output characterisation of the time varying leaky bucket shapers. This is equivalent to computing $\bar{W}$, when W is given by Equation (2). We could try to obtain $\bar{W}$ from a direct application of Equation (1), however this is not a very practical approach. Instead, we obtain $\bar{W}$ from a number of intermediate steps, which provide representations that can easily be applied to a practical computation and give some insights about the system.

To this aim, we first study a shaper system defined by J unchanging leaky buckets, but whose initial conditions (initial bucket levels and initial buffer content) are not zero. We call this model a *leaky bucket shaper with non-zero initial conditions*. We find the input-output characterisation of this model; for this we use min-plus algebra ([12], [13], [6], [11]). Then we apply this iteratively to derive the input characterisation of a time varying leaky bucket shaper (Section 2).

The RVBR Service and its application to RSVP

We derive the input-output characterisation of the RVBR service as a special case of the time varying leaky bucket shaper. An RVBR source is a time varying leaky-bucket shapers with two renegotiable leaky buckets ($J = 2$); one with rate r_i and depth b_i and the second with rate p_i and depth always equal to zero, plus a buffer of fixed size X. In real life, examples of this service are traffic shaping done at source sending over VBR connections as defined in [14] and Internet traffic that takes the form of IntServ specification with RSVP reservation [15], [16]. Indeed, we show that the RVBR service can be used to renegotiate resource reservation for Internet traffic with RSVP. In RSVP the sender sends a PATH message with a *Tspec* object which characterises the traffic it is willing to send. If we consider a network that provides a service as specified for the Controlled Load service (CL) [17], the *Tspec* takes the form of a double bucket specification [18] as given by the RVBR service. There is a peak rate p and a leaky bucket specification with rate r and bucket size b. Additionally there is a minimum policed unit m and a maximum packet size M. We ignore m and M, which are assumed to be fixed. With RSVP as reservation protocol, the reservation has to be periodically refreshed. The suggested period is 30 seconds. Therefore p, r and b need to be reissued at each renegotiation time. There is no additional signaling cost in applying a *Tspec* renegotiation at that point, even if there is some computational overhead due to the computation of the new parameters, or to the call admission control, etc. It is important to note here that, contrary to the negotiation of a new connection, with the renegotiation the reservation is never interrupted.

If the requested traffic specification cannot be supported by the network, the old traffic specification is restored and the network may not be able to accommodate the next traffic. Mechanisms to prevent this failure from occurring are still under study. Here we assume that the *Tspec* is accepted all over the network as well as at the destination, such that the source can transmit conforming to its desired traffic specification.

To apply the RVBR service in this scenario we assume that at any time $t_i = 30 \cdot i$ the application knows (because pre-recorded or predicted) the traffic for the next 30 seconds. We further assume to know the cost to the network of the *Tspec*s (indicated by the cost function $u \cdot r + b$) and the upper bound to the bucket size b_{max} and to the bucket rate r_{max}. The backlog $w(t_i)$ and the bucket level $q(t_i)$ can be measured in the system. Then, with the RVBR service, we compute the *Tspec* that the sender will send at the next renegotiation time. In this context we do not consider delay issues (delay incorporation, as well as the extension to Guaranteed Service [19], is matter of further study).

Previous results and work breakdown

Recent research has introduced an output characterisation of shaper systems in terms of the network calculus theory [20] and [11]. This was used in several papers to characterise the VBR service [13] and [6]. The optimisation problem for the VBR service was studied in [5], [6].

Renegotiation was first specified in ATM networks for CBR class service [21] and only very recently to VBR class service [14]. In the reservation protocol for Integrated Services Internet networks, namely RSVP, a source is requested to refresh the reservation at given times. However, this is not intended as a mechanism for modifying the reservation parameters only, but rather as the general approach for managing the reservation state in routers and hosts [15].

Renegotiable VBR services are also studied in [22],[23],[24]; there the focus is on describing a given traffic with as few leaky buckets as possible, and thus applies to the optimization of a network offering the RVBR service. In contrast, our approach focuses on the customer side of the RVBR service, and provides an analysis of the various tradeoffs that can be made. Our work also differs the systematic use of network calculus. This results in simple, algorithms that can easily be implemented in real applications.

As already mentioned, the parameters optimisation for the RVBR service is not a trivial problem. This problem can be reduced to an optimisation problem by introducing a cost function that associates a cost to each feasible choice of σ_i. We can approach this optimisation problem in different ways. We can minimise the cost of σ_i at each interval I_i given the status of the system at t_i and the input function $R(t)$ in I_i. The result is a sequence of local optimal σ_i. Alternately, we can minimise the cost of the global sequence of σ_i given the complete input function $R(t)$. The result is the optimal sequence of σ_i. The latter, studied in [28], can be seen as a theoretical limit to the previous one and is not presented here.

In the next section, as our first finding, we characterise a leaky-bucket shaper system with non-zero initial conditions in terms of input-output functions. Second, we define the bucket level and the backlog for the time varying leaky-bucket shaper. Hence, by combining these results, we deduce a recursive input-output characterisation of the time varying leaky-bucket shaper. Finally we introduce the RVBR service, which is characterised by using the previous results.

In Section 3, we extend the previous work on the VBR shaper [5], [6] for solving the *local problem*. The cost function is represented by a linear cost function and we are able to provide an algorithm.

As the main application of the RVBR service is the *Tspec* renegotiation, we simulate RVBR in the RSVP with CL service case. We evaluate the effectiveness of the RVBR algorithm for linear cost function (*localOptimum1*) in terms of cost and backlog. We also compare the output of this algorithm with the solution obtained when resetting the buckets at each transient time.

2 Input-Output Characterisation of the Time Varying Leaky Bucket Shaper

The class of leaky-bucket shaper with non-zero initial conditions has the advantage that can easily be studied with network calculus. We derive its input-output characterisation, which can be expressed in terms of the shaping function σ and the initial conditions. Then, in Section 2.2, we derive recursively the characterisation of the time varying leaky-bucket shaper. At the end of the section, we give the input-output characterisation of the RVBR service as special case of the time varying leaky-bucket shaper. The RVBR input-output characterisation will be used in the rest of the paper.

2.1 Leaky-Bucket Shaper with Non-Zero Initial Conditions

The main result in this section is the characterisation of the leaky-bucket shaper with non-zero initial conditions given in Theorem 1. With non-zero initial conditions we refer to the fact that both the buffer and the buckets present an initial level different from zero. The initial bucket level for the j-th bucket is indicated with q_0^j. When a bit enters the system it is put into the bucket, which is drained at rate r^j. A given flow S is conform to a leaky bucket specification when the bucket does not overflow. If we denote with $q(t)$ the bucket level of the bucket at time t, we recall the following characterisation. A flow S is compliant to a leaky bucket with a leaky bucket specification (r, b) when $q(t) \leq b$ $\forall t \geq 0$.

We first present a result that is valid for generic shaper systems.

Proposition 1 (Shaper with non-zero initial buffer) *Consider a shaper system with shaping curve σ. Assume that σ is sub-additive and $\sigma(0) = 0$. Assume the initial buffer content of the shaping buffer is given by w_0. The shaper system has no memory of the past. Then the output R^* for a given input R is*

$$R^*(t) = \sigma(t) \wedge \inf_{0 \leq s \leq t} \{(R)(s) + w_0 + \sigma(t-s)\} \quad \forall t \geq 0 \tag{3}$$

The condition that σ is sub-additive and $\sigma(0) = 0$ is a technical assumption which is not limiting in practice, since any shaping curve can be replaced by a function satisfying the condition [3, 20]. In particular, the shaping functions associated with leaky buckets do satisfy these assumptions.

Proof:
First we derive the constraints on the output of the shaper. σ is the shaping function thus, for all $t \geq s \geq 0$

$$R^*(t) \leq R^*(s) + \sigma(t-s)$$

and given that the bucket at time zero is not empty, for any $t \geq 0$, we have that

$$R^*(t) \leq R(t) + w_0$$

At time $s = 0$, no data has left the system and this can be expressed with the burst delay function δ_0 defined as follow

$$\delta_0(t) = \begin{cases} 0 & t \leq 0 \\ +\infty & t > 0 \end{cases}$$

Thus, for all $t \geq 0$

$$R^*(t) \leq \delta_0(t)$$

The output is thus constrained by

$$R^* \leq \sigma \otimes R^* \wedge R + w_0 \wedge \delta_0$$

where $\otimes$ is the min-plus convolution operation, defined by $(f \otimes g)(t) = \inf_s f(s) + g(t-s)$. Since the shaper is an optimal shaper, the output is the maximum function satisfying this inequality. We know from min-plus algebra [3, 10] that the solution is given by

$$\begin{aligned} R^* &= \sigma \otimes [(R + w_0) \wedge \delta_0] \\ &= [\sigma \otimes (R + w_0)] \wedge [\sigma \otimes \delta_0] \\ &= [\sigma \otimes (R + w_0)] \wedge \sigma \end{aligned}$$

which after some expansion gives the formula in the proposition. □

In practice this proposition says that, whenever a buffer contains some traffic, this has to be considered as a peak arriving at time $t = 0$. The effect of the peak is the factor $\sigma(t)$ in the representation of the output. An easy derivation is the following corollary.

Corollary 1 (Backlog for a shaper with non-zero initial buffer) *The backlog of a flow S into a buffer drained at rate r with initial level equal to L_0 is given by*

$$L(t) = \max \left[\begin{array}{l} \sup_{0 < s \leq t} \{S(t) - S(s) - r \cdot (t-s)\} \\ {[S(t) - r \cdot t + L_0]} \end{array} \right] \quad t \geq 0 \tag{4}$$

Definition 1 *A given traffic S is compliant to the specification of a leaky-bucket shaper system with non-zero initial conditions if it is compliant to all J leaky buckets.*

From Proposition 1 this results in the following corollary.

Corollary 2 (Compliance to J leaky buckets with non-zero initial bucket levels) *A flow S is compliant to J leaky buckets with leaky bucket specifications (r^j, b^j), $j = 1, 2 \ldots J$ and initial bucket level q_0^j iff*

$$\begin{aligned} S(t) - S(s) &\leq \min_{1 \leq j \leq J} [r^j \cdot (t-s) + b^j] \quad \forall 0 < s \leq t \\ S(t) &\leq \min_{1 \leq j \leq J} [r^j \cdot t + b^j - q_0^j] \quad \forall t \geq 0 \end{aligned}$$

Now we proceed to characterise a leaky-bucket shaper system with non-zero initial bucket levels.

Proposition 2 (Leaky-Bucket Shaper with non-zero initial bucket levels) *Consider a shaper system defined by J leaky buckets (r^j, b^j), with $j = 1, 2 \ldots J$ (leaky-bucket shapers). Assume that the initial bucket level of the j-th bucket is given by q_0^j. The initial level of the shaping buffer is equal to zero. The output R^* for a given input R is*

$$R^*(t) = \min[\sigma^0(t), (\sigma \otimes R)(t)] \quad \forall t \geq 0 \tag{5}$$

where σ is the shaping function

$$\sigma(u) = \min_{1 \leq j \leq J} \{\sigma^j(u)\} = \min_{1 \leq j \leq J} \{r^j \cdot u + b^j\}$$

and σ^0 is defined as

$$\sigma^0(u) = \min_{1 \leq j \leq J} \{r^j \cdot u + b^j - q_0^j\}$$

Proof:
The proof, which is not given here, comes by applying to

Corollary 2 the same min-plus result as in Proposition 1. The formal proof is given in [28]. □

Finally we derive the characterisation of a leaky-bucket shaper that starts with non-zero initial conditions.

Theorem 1 (Leaky-Bucket Shaper with non-zero initial conditions)
Consider a shaper system defined by J leaky buckets (r^j, b^j), with $j = 1, 2 \dots J$ (leaky-bucket shaper). Assume that the initial buffer level of the shaping buffer is given by w_0 and the initial bucket level of the j-th bucket is given by q_0^j. The output R^* for a given input R is

$$R^*(t) = \min\{\sigma^0(t), w_0 + \inf_{u>0}\{R(u) + \sigma(t-u)\}\} \quad \forall t \geq 0 \tag{6}$$

with

$$\sigma^0(u) = \min_{1\leq j\leq J}(r^j \cdot u + b^j - q_0^j)$$

Proof:
The proof comes directly from Propositions 1 and 2. □

An intuitive interpretation that generalises Equation (6) is to say that any shaper system starting with non-zero initial conditions offers a service that is either the service offered by an ordinary leaky-bucket shaper, taking into account the initial level of the buffer, or, if smaller, a service imposed by the initial conditions, independently from the input. For the class of the leaky-bucket shaper with non-zero initial conditions, we are also able to define the service imposed by the initial conditions as function of the buckets level.

Example Assume to have a leaky-bucket shaper with non-zero initial conditions defined by 3 leaky buckets: leaky bucket LB1 with $(r_1 = 2, b_1 = 0)$; leaky bucket LB2 with $(r_2 = 1, b_2 = 1)$; leaky bucket LB3 with $(r_3 = \frac{1}{2}, b_3 = 3)$; and a shaping buffer of capacity $X = 4$. Assume the initial conditions are as follows: the level of the bucket LB1 is zero; the level of the bucket LB2 is equal to $\frac{1}{2}$; the level of the bucket LB2 is equal to 1; the initial level of the shaping buffer is $w_0 = 2$. The shaping function σ and the function σ^0 are illustrated in Figure 2(a). Then we analyse the cases of input flows $S1$ and $S2$.

Case 1: In the beginning the amount of traffic issued with $S1$ is not very large and the buckets can handle it without using the buffer anymore, regardless of the initial bucket levels and the initial level of the buffer. Indeed, the quantity of input is smaller than the output, thus the buffer empties. At time $t = 3$ the flow $S1$ arrives with a large amount of traffic. For this reason, after this time, the buckets cannot handle all the traffic and the buffer starts to fill again. At time $t = 6$ the buffer is full. Every time the output coincides with the function σ^0. This case is illustrated in Figure 2(b). With respect to Equation (6), $S1^*$ is computed as $\sigma_0(t)$ for any t. This means that the constraint imposed by the initial conditions is always more strong than the action of the shaping function on $S1$.

Case 2: The flow $S2$ presents always a quantity of traffic that can be absorbed by the leaky buckets without using the buffer, even considering the initial conditions. The output coincides with σ^0 in the beginning and with the flow $(S2 \otimes \sigma) + w_0$ for $t > 11$ to the end. For $t \in [4, 11]$, $S2^* = (\sigma_0)(t)$ for $t \geq 5$. The shaping buffer empties at time $t = 4$, varies for $4 \leq t \leq 11$, empties again at $t = 11$ and remains empty after that time. Figure 2(c) shows $S2$ and $S2^*$. This is an example of a case where the shaping done by σ is sometimes more relevant than the constraint imposed by the initial conditions.

2.2 Time Varying Leaky-Bucket Shaper Model

As introduced in Section 1, at the time instant t_i, where the leaky bucket parameters are changed, we keep the leaky bucket level $q^j(t_i)$ unchanged. We can apply the results in Section 2.1 to a sequence of intervals. We obtain (Proposition 3 of [28]) that the bucket level $q^j(t)$ of the j-th bucket is, for $t \in I_i$

$$q^j(t) = \max\left[\begin{array}{l} \sup_{t_i<s\leq t}\{R^*(t) - R^*(s) - r_i^j \cdot (t-s)\}, \\ \left[R^*(t) - R^*(t_i) - r_i^j \cdot (t - t_i) + q^j(t_i)\right] \end{array}\right] \tag{7}$$

We can now characterise a time varying leaky-bucket shaper in the interval I_i by iterarting Theorem 1. The initial conditions are represented by $q^j(t_i)$ and $w(t_i)$, which are respectively the bucket level and the backlog that are found by the traffic arriving in the interval I_i.

Theorem 2 (Time Varying Leaky-Bucket Shapers)
Consider a time varying leaky-bucket shaper with shaping curve σ_i in the interval I_i. The output R^* for a given input R is

$$R^*(t) = \min\left[\sigma_i^0(t - t_i) + R^*(t_i), \inf_{t_i<s\leq t}\{\sigma_i(t-s) + R(s)\}\right] \tag{8}$$

where σ_i^0 is defined as

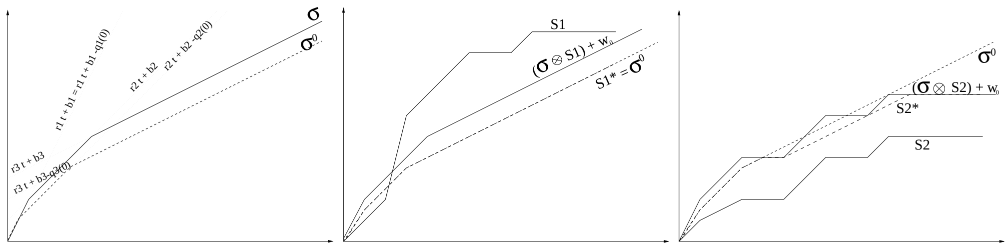
$$\sigma_i^0(u) = \min_{1\leq j\leq J}\left[r_i^j \cdot u + b_i^j - q^j(t_i)\right]$$

The backlog at time t is

$$w(t) = \max\left[\begin{array}{l} \sup_{t_i<s\leq t}\{R(t) - R(s) - \sigma_i(t-s)\}, \\ R(t) - R(t_i) - \sigma_i^0(t - t_i) + w(t_i) \end{array}\right] \quad t \in I_i \tag{9}$$

Proof:
The demonstration is given in [28]. □

In practice, for the class of time varying leaky-bucket



(a) Functions σ and σ^0 (b) Output of S1 (c) Output of S2

Figure 2: Output $S1^*$ and $S2^*$ of a shaper system with non-zero initial conditions for $S1$ and $S2$.

shapers, this theorem gives the closure of W discussed in the introduction. Even this result has an intuitive interpretation that can be generalised for the class of time varying shapers. The output of a time varying shaper in any interval is either driven by σ_0 as combination of the shaping function and the past history, or is computed by taking into account the level of the shaping buffer at the beginning of the interval. This definition is evidently recursive because it depends on the output and on the past history, which are themselves computed with the same formulas. For a discussion on linear time varying shapers see [29].

The definition of the RVBR service comes straightforward as a special case of time varying leaky-bucket shapers, where $J = 2$. Therefore, in the Equations (8) and (9), σ_i and σ_i^0 are given by

$$\sigma_i(u) = \min(p_i \cdot u + b_i^1, r_i \cdot u + b_i^2) \tag{10}$$

$$\sigma_i^0(u) = \min(p_i \cdot u + b_i^1 - q^1(t_i), r_i \cdot u + b_i^2 - q^2(t_i)) \tag{11}$$

In conclusion of this section, we recall that the DGCRA is an example of time varying leaky-bucket shapers. We only mention that the output of a node regulated by the DGCRA is equivalent to the output of a time varying leaky-bucket shaper with $J = 2$. The easy proof is left to the reader.

3 RVBR Service: the Local Optimisation Problems

3.1 Optimisation Algorithm

So far we have assumed that at each interval the bucket specifications are given. In this section, we analyse the problem of computing leaky bucket parameters for the RVBR service, because we want to use RVBR service for RSVP with CL service scenario. Therefore, we study the case of a source that wants to reserve the resources for the next interval. For the RVBR service, this is equivalent to the problem of computing the RVBR parameters for the next interval. In particular, referring to the Equations (10) and (11), b_i^1 is assumed to be fixed and in order to simplify the notation, equal to zero. Therefore we indicate the RVBR parameters at the interval I_i with p_i, r_i and b_i.

We assume that $q^j(t) \leq b_i$ for $t \in I_i$ holds and that we guarantee the service, namely $w(t) \leq X$. From Equation (9) of Proposition 2, we obtain

$$\begin{array}{lr} R(t) - R(s) \leq \sigma_i(t-s) + X & t \in I_i, t_i < s \leq t \\ R(t) - R(t_i) \leq \sigma_i^0(t-t_i) - w(t_i) + X & t \in I_i \end{array}$$

The equations give a necessary and sufficient condition for a minimum p_i

$$p_i = \max \left(\begin{array}{l} \sup\limits_{t,s \in I_i} \dfrac{R(t) - R(s) - X}{t-s}, \\ \sup\limits_{t \in I_i} \dfrac{R(t) - R(t_i) - X + w(t_i)}{t - t_i} \end{array} \right) \tag{12}$$

In analogy to the work in [6] this can be seen as the *effective bandwidth* of the arrival stream in I_i taking in account the backlog at time t_i.

This means that, given that p_i is computed independently from r_i and b_i, the problem of finding a complete optimal parameter set (p_i, r_i, b_i) for the RVBR service is reduced to the problem of finding the optimal parameters r_i and b_i. This is an important aspect of RVBR service. In fact the effective bandwidth p_i is also the minimal peak rate selection for RCBR service. Therefore the two parameters r_i and b_i can only lead to better performance.

We assume that r_i and b_i are limited not to exceed some maximum value that is fixed over time (thus valid for all i), that we indicate with r_{max} and b_{max}.

We define with β_i a function that, for each $s \in I = [0, t_{i+1} - t_i]$, computes the maximum amount of traffic sent over the any interval of size s, taking in account the conditions at time t_i.

$$\beta_i(s) = \max \left(\begin{array}{c} \sup\limits_{0 \leq v \leq t_{i+1} - t_i - s} \{R(v+s) - R(v)\} \\ R(s + t_i) - R(t_i) + w(t_i) + q(t_i) \end{array} \right)$$

When the cost function is linear the optimisation problem is to minimise $c(r_i, b_i) = u \cdot r_i + b_i$, for fixed values of u. Therefore at each interval I_i, our problem is to minimise $u \cdot r_i + b_i$ in the acceptance region defined by

$$\begin{array}{l} 0 \le r_i \le r_{max} \\ 0 \le b_i \le b_{max} \\ b_i + r_i \cdot s + X - \beta_i(s) \ge 0 \quad \forall s \in I \end{array} \tag{13}$$

where $I = [0, t_{i+1} - t_i]$. One important condition that must be respected [5, 13] is

$$b_{max} \ge \sup_{s \in I} \{\beta_i(s) - r_{max} \cdot s - X\} \tag{14}$$

otherwise there are no feasible solutions for r_i and b_i and this must be true at any interval.

As stated in [28] the feasible region can be studied as intersection of two regions. This, given that the cost function is non decreasing, reduces the problem to the problem of finding the optimum on the border of the intersection, delimited by $x_A = \sup_{s \in I, s>0} \frac{\beta_i(s) - X - b_{max}}{s}$ and $x_B = \sup_{s \in I, s>0} \frac{\beta_i(s) - X}{s}$. The optimisation problem becomes

$$\text{minimise } ux - \check{\beta}_i(x) \text{ in the region } x_A \le x \le \min(x_B, r_{max}, p_i) \tag{15}$$

In this problem if u is non-positive the minimisation function is wide-sense decreasing and in this case the solution is given by $\min\{x_B, \min(r_{max}, p_i)\}$. If $u > 0$ and the minimum x_0 of the minimisation function is in the interval $[x_A, \min\{x_B, \min(r_{max}, p_i)\}]$ the optimum is for x_0. In particular, if $\beta_i(\cdot)$ is concave, $x_0 = \sup_{s \in I} \frac{\beta_i(s) - \beta_i(u)}{s - u}$. If x_0 is not feasible for the region defined in Equation (15) we can have $x_0 \le x_A$ and in this case the optimum is found at x_A. Otherwise $x_0 \ge \min(x_B, \min(r_{max}, p_i))$ and therefore the optimum is $\min(x_B, \min(r_{max}, p_i))$.

Finally, we can summarise these results in the algorithm *localOptimum1* that finds the optimal solution as described above. The algorithm is given for $\beta_i(\cdot)$ concave. When this does not hold it is substituted by $\beta_i^!(\cdot)$, as described in [28].

Algorithm 1 localOptimum1

if $b_{max} < \sup_{s \in I}\{\beta_i(s) - r_{max} \cdot s - X\}$ **then** there is no feasible solution;
else {

$$p_i = \max \left(\begin{array}{c} \sup_{t,s \in I_i} \frac{R(t) - R(s) - X}{t - s}, \\ \sup_{t \in I_i} \frac{R(t) - R(t_i) - X + w(t_i)}{t - t_i} \end{array} \right)$$

if $u \le 0$ **then** {
$x_0 = min(r_{max}, p_i)$;
}
else {
$x_0 = \sup_{s \in I} \frac{\beta_i(s) - \beta_i(u)}{s - u}$;
$x_A = \sup_{s \in I, s>0} \frac{\beta_i(s) - X - b_{max}}{s}$;
$x_B = \sup_{s \in I, s>0} \frac{\beta_i(s) - X}{s}$;
if $(x_0 \ge \min(x_B, r_{max}, p_i))$ **then** $x_0 = \min(x_B, r_{max}, p_i)$;
else if $(x_0 \le x_A)$ **then** $x_0 = x_A$;
}
$r_i = x_0$;
$b_i = \sup_{s \in I}\{\beta_i(s) - X - s \cdot x_0\}$;

In [28] we analyse a second version of the optimisation problem, not given here, where the cost of each solution is represented by the reciproc of the number N_i of homogeneous connections, specified by (p_i, b_i, r_i), acceptable by a link with fixed capacity C and buffer with fixed size B .

3.2 Simulation results

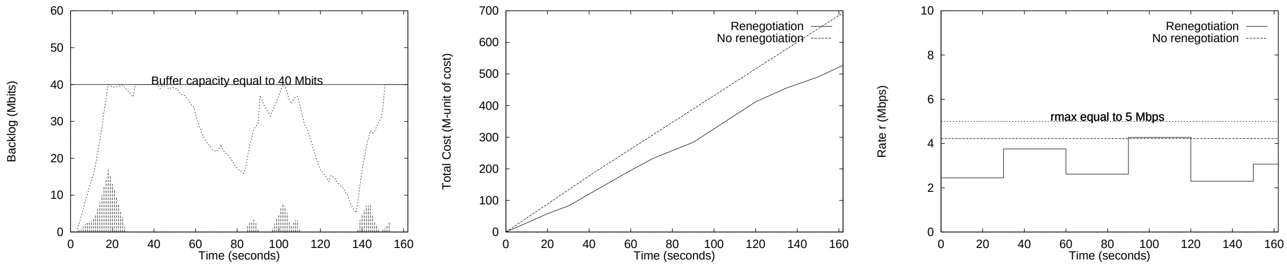
In this section we describe how we use the local algorithm to simulate a typical real case: transmission of MPEG2-encoded video using the IntServ Controlled Load service with the RSVP reservation protocol.

In our simulations, we use a 4000 frame-long sequence that conforms to the ITU-R 601 format ($720 * 576$ at 25 fps). The sequence is composed of several video scenes that differ in terms of spatial and temporal complexities. It has been encoded in an open-loop variable bit rate (OL-VBR) mode, as interlaced video, with a structure of 11 images between each pair of I-pictures and 2 B-pictures between every reference picture. For this purpose, the widely accepted TM5 video encoder [30] has been utilised.

The traffic generated by the video is transported by a trunk regulated by a RVBR service (p, r, b) with shaping buffer X. In this context we do not consider any scheduling issues, which is the subject of ongoing work. Therefore we assume that the video, with a total size of 550 Mbits, is transmitted in 163 seconds (25 frames pro second). The cost function is linear with u. For space reason, we limit to illustrate here only one scenario. Other scenarios are given in [1] and [28]. Here we have that $X = 40$ Mbits, $r_{max} = 5$ Mbps, $b_{max} = 9$ Mbps and $u = 1$. The initial conditions are: $q(0) = 0$ and $w(0) = 0$. The file is pre-recorded and, given that we do not enter in scheduling matters, we know $R(t)$ for all t. At time t_i we know $R^*(t)$ for $t \le t_i$, we measure $w(t_i)$, $q(t_i)$ and compute $\beta_i(t)$. We obtain the optimal shaper parameters by applying the algorithm *localOptimum1* at Section 3 that we use to generate the *Tspec* the sender will send at the next renegotiation time.

In Figure 3(a) we plot the backlog for the scenario in both cases where we apply the renegotiation and where we do not renegotiate [1]. We observe that in the beginning the curves representing the two approaches do not differ much. This is because the traffic is very high in the first 30 seconds and both traffic specifications conform to this traffic.

[1] Even in this case we compute the optimal traffic specification as introduced in [6].



(a) Shaping buffer used withrenegotiation (white area) and without renegotiation (black area)

(b) Cost of allocating a renegotiated traffic specification and a traffic specification without renegotiation

(c) Evolution of the rate r

Figure 3: Comparison between the renegotiation case and the case without renegotiation of the shaping buffer, the cost of the traffic specification and the evolution of the rate r. The cost of the traffic specification is given in "millions of unit of cost" (M-unit of cost).

After that period the traffic rate decreases. The case without renegotiation has to keep the traffic specification negotiated at time $t = 0$, even if it is no longer adequate for the current demand. The resources allocated in the network are so large that it is possible to empty the buffer and thereafter the buffer is rarely used.

The curve for the case where we used the RVBR service shows that the buffer is much better utilised, because the traffic specification decreases in the next intervals.

Therefore, with the RVBR service the resources in the network are much better used. In fact, when the buffer is almost always filled the output is conforms to the traffic specification and this means that all the resources in the network are optimally used. The usage of the buffer with renegotiation is 58%, while without renegotiation it is 13%.

In the graphs in Figure 3(b) we compare the two approaches in terms of the cost of the traffic specification to the network. The cost of the traffic specification is given in terms of the linear cost function used by the RVBR service in order to compute the optimal traffic parameters. The additional result we derive here is that there is also a substantial advantage from the cost point of view in reallocating, because the cost of the traffic specifications is in general smaller.

Figure 3(c) illustrates the fact that with renegotiation we can optimise the resources requested to the network and therefore at the end the total r and b allocated in this case are in general smaller. We also notice that inside an interval the RVBR service might allocate a *Tspec* that is larger than the one used when not renegotiating. This occurs when the traffic is very bursty and the buffer is full from the previous interval, i.e. at the forth interval ($90 - 120$ seconds).

4 "Reset" versus "No Reset" Approach

It is trivial that, in terms of costs, the "reset" approach is better because it always restarts from a zero initial condition and considers the lost traffic as sent.

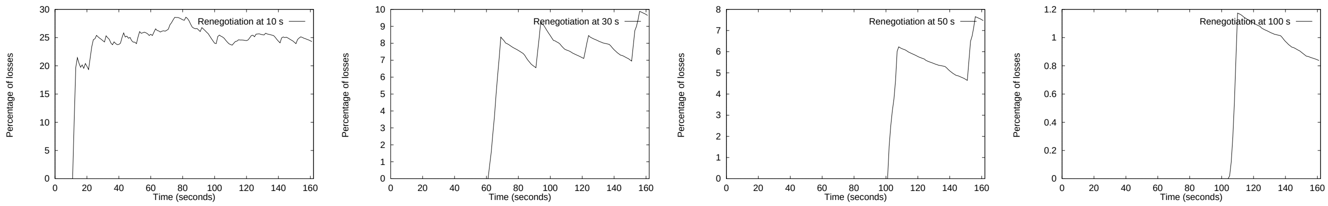
First we point out that the network must use the "no reset" approach because it must ensure to any input traffic, exactly the same service when traffic specification is always renegotiated with the same $\sigma_i = \sigma$ and when the traffic specification is equal to σ and is not renegotiated. This is not possible if the network resets the buckets level at every renegotiation time.

In principle, at the source both approaches are valid. When we reset the buckets we must accept to experience some loss due to the fact that the network does not apply any reset. This means that the upper bound to those losses is given by the maximum size of the bucket (b_{max}) times the number of times we apply the renegotiation. Therefore an upper bound for the percentage of losses is given by

$$\min_i \frac{b_{max} \cdot i}{R(t_i)} \tag{16}$$

It is already clear that this upper bound can be not acceptable for many types of traffic. In practice this limit is easily reached, unless b_{max} is very small. Only in this case, where we have that $\frac{b_{max}}{R(t_i)}$ is close to zero for any value of i, the impact of the reset does not affect the system behaviour. Evidently we can assume that this condition should not occur, because it corresponds to a bad network planning.

To evaluate how close we get to this upper bound we simulate the two approaches in the same scenario described in Section 3.2, where we use IntServ services with RSVP reservation protocol. We measure the percentage of losses, that obviously depends on b_{max}. For the renegotiation at every 30 seconds, we experience a percentage of losses from



(a) renegotiation every 10 seconds

(b) renegotiation every 30 seconds

(c) renegotiation every 50 seconds

(d) renegotiation every 100 seconds

Figure 4: Percentage of losses in the reset approach

5%, for b_{max} very small, up to 60%. Obviously, for a fixed b_{max}, the percentage of losses grows with the decrease of the renegotiation period. For very small renegotiation periods can be enormous.

In Figures 4 we illustrate the losses for an average b_{max} (compared to the input traffic, $b_{max} = 6$ Mbits) for different renegotiation periods. We observe that for most of these cases the percentage of losses is not acceptable. It is different in the case of renegotiation at 100 seconds because here the renegotiation is quite infrequent.

5 Conclusion

For the class of time varying leaky-bucket shapers we have found an explicit representation of the output in terms of the input function (input-output characterisation). This is obtained by iterating the input-output characterisation we derive for the class of leaky-bucket shapers with non-zero initial conditions.

Then we use this result to study the local optimisation aspects of the RVBR service, leaving aside the problem of traffic prediction. The solution is the Algorithms 1 (Section 3).

Furthermore we illustrate how the RVBR service can be applied to RSVP *Path* message generation. This is based on the algorithm proposed for the local optimisation problem. A numerical example of this is given in Section 3.2, where we also compare the performance of transmitting a MPEG2 video trace both with and without renegotiation. The results of our simulation (see Figure 3) suggest that renegotiation allows to better use of network resources and that in protocols as RSVP, where there is no additional cost for signaling (or so we mainly assume), it is better to renegotiate. Future work on RVBR service includes both the possible integration in a real application and study on the renegotiation period, as well as the integration of the network delay and the application to Guaranteed Service [19].

We have also illustrated that, if some inconsistency exists between network and user sides about the use of the "reset" or "no-reset" approach, then this may result in inacceptable losses (or service degradation) due to policing. We give an upper bound to the percentage of losses and we notice that in general this upper bound is not acceptable, especially for small renegotiation periods. We also found, in the cases we analysed, that this limit can be easily approached. Some simulation results are given in Figure 4 in Section 4.

The results we obtained shows that the RVBR service can be easily and efficiently adopted by video applications requiring strict guaranteed service. In further work our results for the class of time varying leaky-bucket shapers will be used to model network resources renegotiation in other scenarios, as, for instance, in the video smoothing case [31].

In [28] we cover some other related issue like the comparison of the local approach against a global approach based on Viterbi-like algorith to give a measure of the optimality of the former in terms of cost, as well as the comparison of renegotiable VBR service against a renegotiable CBR service. We also discuss certain aspects related to the definition of the optimal renegotiation period.

References

[1] S. Giordano, J.-Y. Le Boudec, "QoS based Integration of IP and ATM: Resource Renegotiation," *In Proceedings of 13th IEEE Computer Communications Workshop*, 1998. http://lrcwww.epfl.ch/~giordano/publications.html.

[2] R. Guérin and V. Peris, "Quality-of-service in packet networks - basic mechanisms and directions," *Computer Networks and ISDN, Special issue on multimedia communications over packet-based networks*, 1998.

[3] C. Chang, "On deterministic traffic regulation and service guarantee: A systematic approach by filtering," *IEEE Transactions on Information Theory*, vol. 44, pp. 1096–1107, August 1998.

[4] A. Ziedinsh and J.-Y. Le Boudec, "Adaptive CAC Algoritms," *in proceedings of ITC 15*, 1996.

[5] S. Giordano, J-Y. Le Boudec, P. Oechslin, S. Robert, "VBR over VBR: the Homogeneous, Loss-free Case," *INFOCOM97*, 1997. http://lrcwww.epfl.ch/~giordano/publications.html.

[6] J.-Y. Le Boudec, "Network Calculus, Deterministic Effective Bandwidth, VBR trunks," *IEEE Globecom 97*, November 1997.

[7] The ATM Forum, *ATM User-Network Interface (UNI) Signalling Specification, Version 4.0*, 1996. ftp://ftp.atmforum.com/pub/approved-specs/af-sig-0061.000.ps.

[8] W. P. 4, "Specification of Integrated Traffic Control Architecture," Deliverable Del06, ACTS project AC094 EXPERT, September 1997.

[9] C. Chang and R. L. Cruz, "A time varying filtering theory for constrained traffic regulation and dynamic service guarantees," in *Prepring*, July 1998.

[10] F. Baccelli, G. Cohen, G. J. Olsderand, and J.-P. Quadrat, *Synchronization and Linearity, An Algebra for Discrete Event Systems.* John Wiley and Sons, 1992.

[11] J.-Y. Le Boudec and P. Thiran, "Network Calculus viewed as a Min-plus System Theory applied to Communication Networks," Technical Report 98/276, DI-EPFL, CH-1015 Lausanne, Switzerland, April 1998.

[12] R.L. Cruz, "Quality of Service Guarantees in Virtual Circuit Switched Networks," *JSAC, August 1995*, 1995.

[13] J.-Y. Le Boudec, "Application of Network Calculus To Guaranteed Service," Technical Report 97/251, DI-EPFL, CH-1015 Lausanne, Switzerland, November 1997.

[14] ITU Telecommunication Standardization Sector - Study group 13, *ITU-T Recommandation Q.2963.2. Broadband integrated services digital network (B-ISDN) digital subscriber signalling system No. 2 (DSS 2) connection modification - Modification procedure for sustainable cell rate parameters*, 1998.

[15] R. Braden, L. Zhang, S. Berson, S. Herzog, S. Jamin, *RFC2205: Resource ReSerVation Protocol (RSVP) - Version 1 Functional Specification.* IETF, September 1997.

[16] J. Wroclawski, *RFC2210: The Use of RSVP with IETF Integrated Services.* IETF, September 1997.

[17] J. Wroclawski, *RFC2211: Specification of Controlled-Load Network Element Service.* IETF, September 1997.

[18] S. Shenker, J. Wroclawski, *RFC2216: Network Element Service Specification Template.* IETF, September 1997.

[19] S. Shenker, C. Partridge, R. Guérin, *RFC2212: Specification of Guaranteed Quality of Service.* IETF, September 1997.

[20] J.-Y. Le Boudec, "Network Calculus Made Easy," Technical Report 96/218, DI-EPFL, CH-1015 Lausanne, Switzerland, December 1996.

[21] ITU Telecommunication Standardization Sector - Study group 13, *ITU-T Recommandation Q.2963.1. : Peak cell rate modification by the connection owner*, 1996.

[22] H. Zhang, E. Knightly, "A New Approach to Support Delay-Sentive VBR Video in Packet-Switching Networks.," *In Proceedings 5th Workshop on Network Operating System Support for Digital Audio and Video (NOSSDAV)*, April 1995.

[23] H. Zhang, E. Knightly, "RED-VBR: A Renegotiation-Based Approach to Support Delay-Sensitive VBR Video," *ACM Multimedia Systems Journal*, 1997.

[24] J. Liebeherr, D. Wrege, "An Efficient Solution to Traffic Characterization of VBR Video in Quality-of-Service Network," *to appear in ACM/Springer Multimedia Systems Journal*, 1998.

[25] M. Grossglauser, "Controle des ressources de resaux sur des echelles temporelles multiples," *Ph.D. Thesis*, 1998.

[26] W. Almesberger, L. Chandran, S. Giordano, J.-Y. Le Boudec, R. Schmid, "Using Quality of Service can be simple: Arequipa with Renegotiable ATM connections," *to appear in: Computer Networks and ISDN Systems.* http://lrcwww.epfl.ch/~giordano/publications.html.

[27] C.-Y. Hsu, A. Ortega, "Joint Selection of Source, Channnel Rate for VBR Video Transmission under ATM Policing Constraints," *IEEE Journal on Selected Areas in Communications*, 1997.

[28] S. Giordano, J.-Y. Le Boudec, "On a Class of Time Varying Shapers with Application to the Renegotiable Variable Bit Rate Service," Technical Report SSC/1998/035, DI-EPFL, CH-1015 Lausanne, Switzerland, 1998. http://lrcwww.epfl.ch/~giordano/publications.html.

[29] R. Agrawal, R.L. Cruz, C.M. Okino, R. Rajan, "A Framework for Adaptive Service Guarantees," *Proceedings of Allerton Conf, Monticello*, 1998.

[30] C. Fogg, "mpeg2encode/mpeg2decode," *MPEG Software Simulation Group*, 1996.

[31] J.-Y. Le Boudec and O. Verscheure, "Optimal Smoothing for Guaranteed Service," Technical Report SSC/98/032, EPFL, October 1997.